Paraphrase Identification Using Weighted Dependencies and Word Semantics

Mihai Lintean and **Vasile Rus**
Department of Computer Science
The University of Memphis
Institute for Intelligent Systems
Memphis, TN 38152, USA
M.Lintean@memphis.edu
vrus@memphis.edu

Abstract

In this paper we propose a novel approach to the task of paraphrase identification. The proposed approach quantifies both the similarity and dissimilarity between two sentences. The similarity and dissimilarity is assessed based on lexico-semantic information, i.e., word semantics, and syntactic information in the form of dependencies, which are explicit syntactic relations between words in a sentence. Word semantics requires mapping words onto concepts in a taxonomy and then using word-to-word similarity metrics to compute their semantic relatedness. Dependencies are obtained using state-of-the-art dependency parsers. One important aspect of our approach is the weighting of missing dependencies, i.e., syntactic relations present in one sentence but not the other. We report experimental results on the Microsoft Paraphrase Corpus, a standard data set for evaluating approaches to paraphrase identification. The experiments showed that the proposed approach offers state-of-the-art results. In particular, our approach offers better precision when compared to other state-of-the-art systems.

Introduction

Paraphrasing is a text-to-text relation between two non-identical text fragments that express the same idea in different ways. As an example of a paraphrase we show below a pair of sentences from the Microsoft Research (MSR) Paraphrase Corpus (Dolan, Quirk, and Brockett 2004) in which Text A is a paraphrase of Text B and vice versa.

Text A: *York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.*

Text B: *York had no problem with MTA's saying the decision to shift funds was within its powers.*

Paraphrase identification is the task of deciding, given two text fragments, whether they have the same meaning. We focus in this paper on identifying paraphrase relations between sentences such as the ones shown above. It should be noted that paraphrase identification is different from paraphrase extraction. Paraphrase extraction (Barzilay and Lee 2003; Brockett and Dolan 2005) is the task of extracting fragments of texts that are in a paraphrase relation from various sources.

Paraphrase identification and extraction are important tasks in a number of applications including Question Answering (Ibrahim, Katz, and Lin 2003), Natural Language Generation (Iordanskaja, Kittredge, and Polgere 1991), and Intelligent Tutoring Systems (Graesser et al. 2005; McNamara et al. 2007). We propose in this paper a fully automated approach to the task of paraphrase identification. The basic idea is that two sentences are in a paraphrase relation if they have many similarities (at lexico-semantic and syntactic levels) and few or no dissimilarities. For instance, the two sentences shown earlier from the MSR paraphrase corpus have many similarities, e.g., common words such as *York* and common syntactic relations such as the *subject* relationship between *York* and *have*, and only a few dissimilarities, e.g., Text A contains the word *saying* while Text B contains the word *insisting*. Thus, we can confidently deem the two sentences as being paraphrases of each other. Following this basic idea, in our approach to paraphrase identification we first compute two scores: one reflecting the similarity and the other the dissimilarity between the two sentences. A paraphrase score is generated by taking the ratio of the similarity and dissimilarity scores. If the ratio is above a certain threshold, the two sentences are judged as being paraphrases of each other. The threshold is obtained by optimizing the performance of the proposed approach on the training data.

There are several key features of our approach that distinguish it from other approaches to paraphrase identification. *First*, it considers both similarities and dissimilarities between sentences. This is an advantage over approaches that only consider the degree of similarity (Rus et al. 2008) because the dissimilarity of two sentences can be very important to identifying paraphrasing, as shown by (Qiu, Kan, and Chua 2006) and later in this paper. *Second*, the similarity between sentences is computed using word-to-word similarity metrics instead of simple word matching or synonymy information in a thesaurus as in (Rus et al. 2008; Qiu, Kan, and Chua 2006). The word-to-word similarity metrics can identify semantically related words even if the words are not identical or synonyms. We use the WordNet similarity metrics (Patwardhan, Banerjee, and Pedersen 2003) which rely on statistical information derived from corpora and lexico-semantic information from WordNet (Miller 1995), a lexical database of English. The gist of the WordNet similarity measures is that the closer the distance is in

WordNet between words/concepts, the more similar they are. For instance, in the earlier example the semantic relationship between the words *insist* and *say* cannot be established using simple direct matching or synonymy. On the other hand, there is a relatively short path of three nodes in WordNet from *say* to *insist* via *assert*, indicating *say* and *insist* are semantically close. *Third*, we weight dependencies to compute dissimilarities between sentences as opposed to simple dependency overlap methods that do no weighting [see (Lintean, Rus, and Graesser 2008; Rus, McCarthy et al. 2008)]. The weighting allows us to make fine distinctions between sentences with a high similarity score that are paraphrases and those that are not due to the strength of the few dissimilarities. For instance, two sentences that are almost identical except their subject relations are likely to be non-paraphrases as opposed to two highly similar sentences that differ in terms of, say, determiner relations. We weight dependencies using two features: (1) the type/label of the dependency, and (2) the depth of a dependency in the dependency tree. To extract dependency information we used two parsers, Minipar (Lin 1993) and the Stanford parser (Maneffe, MacCartney, and Manning 1991). We report results with each of the parsers.

We used the MSR Paraphase Corpus (Dolan, Quirk, and Brockett 2004), an industry standard for paraphrase identification in Natural Language Processing, to evaluate our approach. The corpus is divided into two subsets: training and test data. The training subset was used to obtain the optimal threshold above which a similarity/dissimilarity ratio would indicate a paraphrase or a non-paraphrase otherwise. We report state-of-the-art results on the testing data (72.06% accuracy, with Minipar), which are significantly better (Fisher's exact test yields a $p = 0.00005$) than the baseline approach of always predicting the most frequent class in the training data (66.49% accuracy) and than a simple dependency overlap method ($p<0.001$; with Minipar). Compared to results obtained using the Stanford parser (71.01% accuracy), Minipar led to statistically significant better results ($p = 0.004$).

The rest of the paper starts with the *Related Work* section. The *Approach* section describes in detail how our similarity-dissimilarity method works. The following *Summary of Results* section provides details of the experimental setup, results, and a comparison with results obtained by other research groups. The *Discussion* section offers further insights into our approach and the MSR Paraphrase Corpus. The *Summary and Conclusions* section ends the paper.

Related Work

Paraphrase identification has been explored in the past by many researchers, especially after the release of the MSR Paraphrase Corpus (Dolan, Quirk, and Brockett 2004). The task of paraphrase identification is related to the task of recognizing textual entailment, which we do not discuss here, due to space constraints. We briefly describe three previous studies that are most related to our approach and leave others out, e.g., (Kozareva and Montoyo 2006; Wu 2005).

Rus and colleagues (Rus et al. 2008) addressed the task of paraphrase identification by computing the degree of subsumption at lexical and syntactic level between two sentences in a bidirectional manner: from Text A to Text B and from Text B to Text A. The approach relied on a unidirectional approach that was initially developed to recognize the sentence-to-sentence relation of entailment (Rus, McCarthy et al. 2008). Rus and colleagues' approach only used similarity to decide paraphrasing, simply discarding dissimilarities without carefully analyzing their importance to the final decision. The similarity was computed as a weighted sum of lexical matching, i.e. direct matching of words enhanced with synonymy information from WordNet, and syntactic matching, i.e., dependency overlap. Dependencies were derived from a phrase-based parser which outputs the major phrases in a sentence and organizes them hierarchically into a parse tree. Our approach has a better lexical component based on word semantics and a finer syntactic analysis component based on weighted dependencies. Furthermore, the use of phrase-based parsing in (Rus et al. 2008) limits the applicability of the approach to free-order languages for which dependency parsing is more suitable.

Corley and Mihalcea (2005) proposed an algorithm that extends word-to-word similarity metrics into a text-to-text semantic similarity metric based on which they decide whether two sentences are paraphrases or not. To get the semantic similarity between words they used the same WordNet similarity package as we do. Our approach has the advantage that it considers syntactic information, in addition to word semantics, to identify paraphrases.

Qiu and colleagues (2006) proposed a two-phase architecture for paraphrase identification. In the first phase, they identified similarities between two sentences, while in the second phase the dissimilarities were classified as to their relevance in making the decision of whether the sentences are paraphrases. Their approach uses predicate argument tuples that capture both lexical and syntactic dependencies among words to find similarities between sentences. The first phase is similar to our approach for detecting common dependencies. In the second phase, they used a supervised classifier to detect whether the dissimilarities are important. There are two advantages of our approach compared to Qiu and colleagues' approach (1) we use word semantics to compute similarities, (2) we take advantage of the dependency types and position in the dependency tree to weight dependencies as opposed to simply using non-weighted/unlabeled predicate-argument relations.

Approach

As mentioned earlier, our approach is based on the observation that two sentences express the same meaning, i.e., are paraphrases, if they have many words and syntactic relations in common. Furthermore, the two sentences should have few or no dissimilar words or syntactic relations. In the example below, we show two sentences with a high lexical and syntactic overlap. The different information, *legal rights* in the first sentence and *powers* in the second sentence, does not have a significant impact on the overall decision that the two sentences are paraphrases based on their high degree of similarity.

Text A: *The decision was within its legal rights.*

Text B: *The decision was within its powers.*

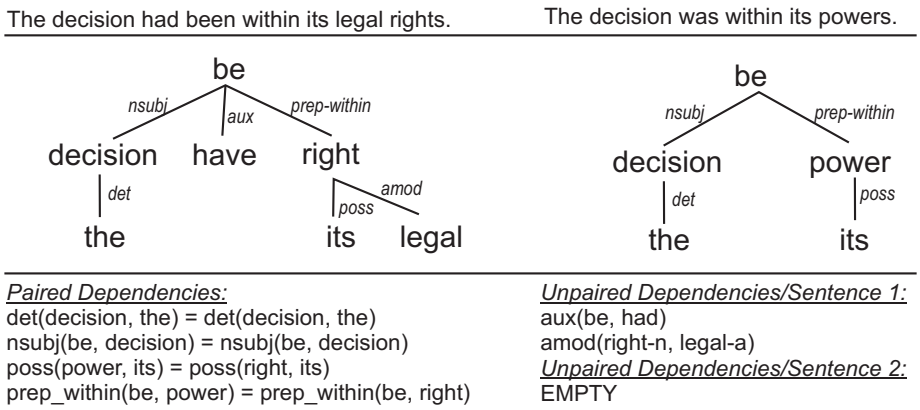

Figure 1: Example of dependency trees and sets of paired and non-paired dependencies.

On the other hand, there are sentences that are almost identical, lexically and syntactically, and yet they are not paraphrases because the few dissimilarities make a big difference. In the example below, there is a relatively "small" difference between the two sentences. Only the subject of the sentences is different. However, due to the importance of the subject relation to the meaning of any sentence the high similarity between the sentences is sufficiently dominated by the "small" dissimilarity to make the two sentences non-paraphrases.

Text A: *CBS is the leader in the 18 to 46 age group.*

Text B: *NBC is the leader in the 18 to 46 age group.*

Thus, it is important to assess both similarities and dissimilarities between two sentences S_1 and S_2 before making a decision with respect to them being paraphrases or not. In our approach, we capture the two aspects, similarity or dissimilarity, and then find the dominant aspect by computing a final paraphrase score as the ratio of the similarity and dissimilarity scores: Paraphrase(S_1, S_2)=Sim(S_1, S_2)/Diss(S_1, S_2). If the paraphrase score is above a learned threshold T the sentences are deemed paraphrases. Otherwise, they are non-paraphrases.

The similarity and dissimilarity scores are computed based on dependency relations (Hays 1964), which are asymmetric relationships between two words in a sentence, a *head*, or modifee, and a *modifier*. A sentence can be represented by a set of dependency relations (see the bottom half of Figure 1). An example of dependency is the *subject* relation between *John* and *drives* in the sentence *John drives a car.* Such a dependency can be viewed as the triple *subj(John, drive)*. In the triplets the words are lemmatized, i.e., all morphological variations of a word are mapped onto its base form. For instance, *go, went, gone, going* are all mapped onto *go*.

The Sim(S_1, S_2) and Diss(S_1,S_2) scores are computed in three phases: (1) map the input sentences into sets of dependencies, (2) detect common and non-common dependencies between the sentences, and (3) compute the Sim(S_1, S_2) and Diss(S_1,S_2) scores. Figure 2 depicts the general architecture of the system in which the three processing phases are shown as the three major modules.

In the first phase, the set of dependencies for the two sentences is extracted using a dependency parser. We use both Minipar (Lin 1993) and the Stanford parser (Maneffe, MacCartney, and Manning 1991) to parse the sentences. Because these parsers do not produce perfect output the reader should regard our results as a lower bound, i.e. results in the presence of parsing errors. Should the parsing been perfect, we expect our results to look better. The parser takes as input the raw sentence and returns as output a dependency tree (Minipar) or a list of dependencies (Stanford). In a dependency tree, every word in the sentence is a modifier of exactly one word, its head, except the head word of the sentence, which does not have a head. The head word of the sentence is the root node in the dependency tree. Given a dependency tree, the list of dependencies can be easily derived by traversing the tree and for each internal node, which is head of at least one dependency, we retrieve triplets of the form *rel(head, modifier)* where *rel* represents the type of dependency that links the node, i.e., the *head*, to one of its children, the *modifier*. Figure 1 shows the set of dependencies in the form of triplets for the dependency trees in the top half of the figure.

In this phase, we also gather positional information about each dependency in the dependency tree as we will need this information later when weighting dependencies in Phase 3. The position/depth of a dependency within the dependency tree is calculated as the distance from the root of the node corresponding to the head word of the dependency. Because the Stanford parser does not provide the position of the dependencies within the tree, we had to recursively reconstruct the tree based on the given set of dependency relations and calculate the relative position of each relation from the root.

The second phase in our approach identifies the common and non-common dependencies of the sentences, based on word semantics and syntactic information. Three sets of dependencies are generated in this phase: one set of *paired*/common dependencies and two sets of *unpaired* dependencies, one corresponding to each of the two sentences. To generate the paired and unpaired sets a two-step procedure is used. In the first step, we take one dependency from the shorter sentence in terms of number of dependencies (a computational efficiency trick) and identify dependencies of

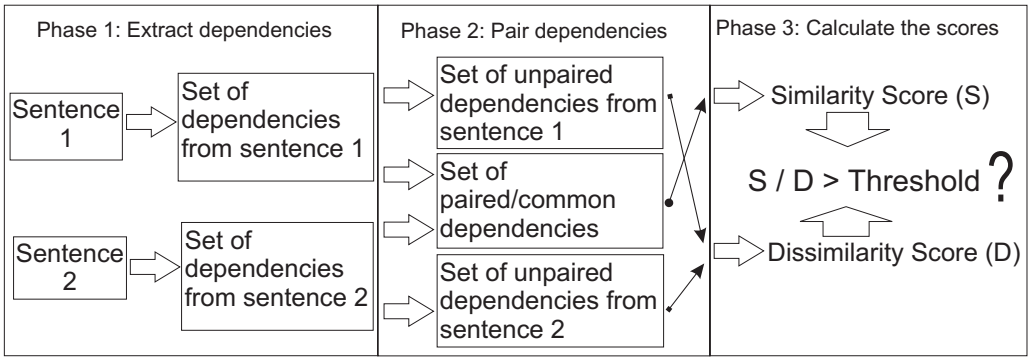

Figure 2: Architecture of the system.

the same type in the other sentence. In the second step, we compute a dependency similarity score (d2dSim) using the word-to-word similarity metrics applied to the two heads and two modifiers of the matched dependencies. Heads and modifiers are mapped onto all the corresponding concepts in WordNet, one concept for each sense of the heads and modifiers. The similarity is computed among all senses/concepts of the two heads and modifiers, respectively, and then the maximum similarity is retained. If a word is not present in WordNet exact matching is used. The word-to-word similarity scores are combined into one final dependency-to-dependency similarity score by taking the weighted average of the similarities of the heads and modifiers. Intuitively, more weight should be given to the similarity score of heads and less to the similarity score of modifiers because heads are the more important words. Surprisingly, while trying to learn a good weighting scheme from the training data we found that the opposite should be applied: more weight should be given to modifiers (0.55) and less to heads (0.45). We believe this is true only for the MSR Paraphrase Corpus and this weighting scheme should not be generalized to other paraphrase corpora. The MSR corpus was built in such a way that favored highly similar sentences in terms of major content words (common or proper nouns) because the extraction of the sentences was based on keyword searching of major events from the web. With the major content words similar, the modifiers are the heavy lifters when it comes to distinguishing between paraphrase and non-paraphrase cases. The dependency-to-dependency similarity score needs to exceed a certain threshold for two matched dependencies to be deemed similar. Empirically, we found out from training data that a good value for this threshold would be *0.5*. Once a pair of dependencies is deemed similar, we place it into the paired dependencies set, along with the calculated dependency-to-dependency similarity value. All the dependencies that could not be paired are moved into the unpaired dependencies sets.

$$sim(S_1, S_2) = \sum_{d_1 \in S_1} max_{d_2 \in S_2^*}[d2dSim(d_1, d_2)]$$

$$diss(S_1, S_2) = \sum_{d_1 \in unpS_1} weight(d_1) + \sum_{d_2 \in unpS_2} weight(d_2)$$

In the third and final phase of our approach, two scores are calculated from the three dependency sets obtained in Phase 2: a cumulative *similarity score* and a cumulative *dissimilarity score*. The cumulative similarity score Sim(S_1, S_2) is computed from the set of paired dependencies by summing up the dependency-to-dependency similarity scores (S_2^* in the equation for similarity score represents the set of remaining unpaired dependencies in the second sentence). Similarly, the dissimilarity score Diss(S_1, S_2) is calculated from the two sets of unpaired dependencies. Each unpaired dependency is weighted based on two features: the depth of the dependency within the dependency tree and type of dependency. The depth is important because an unpaired dependency that is closer to the root of the dependency tree, e.g., the main verb/predicate of sentence, is more important to indicate a big difference between two sentences. In our approach, each unpaired dependency is initially given a perfect weight of 1.00, which is then gradually penalized with a constant value (*0.20* for the Minipar output and *0.18* for the Stanford output), the farther away it is from the root node. The penalty values were derived empirically from training data. Our tests show that this particular feature works well only when applied to the sets of unpaired dependencies. The second feature that we use to weight dependencies is the type of dependency. For example a *subj* dependency, which is the relation between the verb and its subject, is more important to decide paraphrasing than a *det* dependency, which is the relation between a noun and its determiner. Each dependency type is assigned an importance level between 0 (no importance) and 1 (maximum importance). The importance level for each dependency type has been established by the authors based on their linguistic knowledge and an analysis of the role of various dependency types in a subset of sentences from the training data.

Once the Sim(S_1, S_2) and Diss(S_1, S_2) scores are available, the paraphrase score is calculated by taking the ratio between the similarity score, S, and the disimilarity score, D, and compare it to the optimum threshold T learned from training data. Formally, if $S/D > T$ then the instance is classified as paraphrase, otherwise is a non-paraphrase. To avoid division by zero for cases in which the two sentences are identical ($D = 0$) the actual implementation tests for

Table 1: Performance and comparison of different approaches on the MS Paraphrase Corpus.

System	Accuracy	Precision	Recall	F-measure
Uniform baseline	0.6649	0.6649	1.0000	0.7987
Random baseline (Corley and Mihalcea 2005)	0.5130	0.6830	0.5000	0.5780
Corley and Mihalcea (2005)	0.7150	0.7230	0.9250	0.8120
Qiu (Qiu, Kan, and Chua 2006)	0.7200	0.7250	0.9340	0.8160
Rus - average (Rus et al. 2008)	0.7061	0.7207	0.9111	0.8048
Simple dependency overlap (Minipar) (Lintean, Rus, and Graesser 2008)	0.6939	0.7109	0.9093	0.7979
Simple dependency overlap (Stanford) (Lintean, Rus, and Graesser 2008)	0.6823	0.7064	0.8936	0.7890
Optimum results (Minipar)	0.7206	0.7404	0.8928	0.8095
Optimum results (Stanford)	0.7101	0.7270	0.9032	0.8056
No word semantics (Minipar)	0.7038	0.7184	0.9119	0.8037
No word semantics (Stanford)	0.7032	0.7237	0.8954	0.8005
No dependency weighting (Minipar)	0.7177	0.7378	0.8928	0.8079
No dependency weighting (Stanford)	0.7067	0.7265	0.8963	0.8025

$S > T * D$. To find the optimum threshold, we did an exhaustive search on the training data set, looking for the value which led to optimum accuracy. This is similar to the sigmoid function of the simple voted perceptron learning algorithm used in (Corley and Mihalcea 2005).

Summary of Results

We experimented with our approach on the MSR Paraphrase Corpus (Dolan, Quirk, and Brockett 2004). The MSR Paraphrase Corpus is the largest publicly available annotated paraphrase corpus which has been used in most of the recent studies that addressed the problem of paraphrase identification. The corpus consists of 5801 sentence pairs collected from newswire articles, 3900 of which were labeled as paraphrases by human annotators. The whole set is divided into a training subset (4076 sentences of which 2753 are true paraphrases) which we have used to determine the optimum threshold T, and a test subset (1725 pairs of which 1147 are true paraphrases) that is used to report the performance results. We report results using four performance metrics: accuracy (percentage of instances correctly predicted out of all instances), precision (percentage of predicted paraphrases that are indeed paraphrases), recall (percentage of true paraphrases that were predicted as such), and f-measure (harmonic mean of precision and recall).

In Table 1 two baselines are reported: a uniform baseline in which the majority class (paraphrase) in the training data is always chosen and a random baseline taken from (Corley and Mihalcea 2005). We next show the results of others including results obtained using the simple dependency overlap method in (Lintean, Rus, and Graesser 2008). The simple dependency overlap method computes the percentage of common dependencies out of the union (divided by two) of the dependencies in the two sentences. Our results are presented in the following order: our best/state-of-the-art system, that uses both word semantics and weighted dependencies, then a version of the proposed approach without word semantics (similarity in this case is 1 if words are identical, case insensitive, or 0 otherwise) and without weighted dependencies, respectively. The conclusion based on our best approach is that a mix of word semantics and weighted dependencies leads to better accuracy and in particular better precision. The best approach leads to significantly better results than the baselines and the simple dependency overlap ($p<0.001$ for the version with Minipar). The comparison between our best results and the results reported by (Corley and Mihalcea 2005) and (Lintean, Rus, and Graesser 2008) is of particular importance. These comparisons indicate that weighted dependencies and word semantics leads to better accuracy and precision than using only word semantics (Corley and Mihalcea 2005) or only simple dependency overlap (Lintean, Rus, and Graesser 2008).

All results in Table 1 were obtained with the *lin* measure from the WordNet similarity package, except the case that did not use WordNet similarity measures at all – the *No word semantics* row. This *lin* measure consistently led to the best performance in our experiments when compared to all the other measures offered by the WordNet similarity package.

For reference, we report in Table 2 results obtained with an optimum threshold calculated from the *test data set*. We deem these results as one type of benchmark results for approaches that rely on WordNet similarity measures and dependencies as they were obtained by optimizing the approach on the testing data. As we can see from the table, the results are not much higher than the results in Table 1 where the threshold was derived from training data.

Table 2: Accuracy results for optimum test threshold values.

Metric	Acc.	Prec.	Rec.	F
Minipar	0.7241	0.7395	0.9032	0.8132
Stanford	0.7130	0.7387	0.8797	0.8030

Discussion

One item worth discussing is the annotation of the MSR Paraphrase Corpus. Some sentences are intentionally labeled as paraphrases in the corpus even when the small dissimilarities are extremely important, e.g. different numbers. Below is a pair of sentences from the corpus in which the "small" difference in both the numbers and the anonymous *stocks* in Text A are not considered important enough for the annotators to judge the two sentences as non-paraphrases.

Text A: *The stock rose $2.11, or about 11 percent, to close on Friday at $21.51 on the New York Stock Exchange.*

Text B: *PG&E Corp. shares jumped $1.63 or 8 percent to $21.03 on the New York Stock Exchange on Friday.*

This makes the corpus more challenging and the fully-automated solutions look worse than they actually are.

Another item worth discussing is the comparison of the dependency parsers. Our experimental results show that Minipar consistently outperforms Stanford, in terms of accuracy of our paraphrase identification approach. Minipar is also faster than Stanford, which first generates the phrase-based syntactic tree for a sentence and then extracts the corresponding sets of dependencies from the phrase-based syntactic tree. For instance, Minipar can parse 1725 pairs of sentences, i.e. 3450 sentences, in 48 seconds while Stanford parser takes 1926 seconds, i.e. 32 minutes and 6 seconds.

Summary and Conclusions

In this paper, we presented a novel approach to solve the problem of paraphrase identification. The approach uses word semantics and weighted dependencies to compute degrees of similarity at word/concept level and at syntactic level between two sentences being judged as being paraphrases or not. The proposed approach offers state of the art performance. In particular, the approach offers high precision due to the use of syntactic information.

References

Barzilay, R., and Lee, L. 2003. Learning to Paraphrase: An Unsupervised Approach Using Multiple Sequence Alignment. In *Proceedings of NAACL 2003*.

Brockett, C., and Dolan, W. B. 2005. Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of the 3rd International Workshop on Paraphrasing*.

Corley, C., and Mihalcea, R. 2005. Measuring the Semantic Similarity of Texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*. Ann Arbor, MI.

Dolan, B.; Quirk, C.; and Brockett, C. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of COLING*, Geneva, Switzerland.

Graesser, A.C.; Olney,A.; Haynes, B.; and Chipman, P. 2005. *Cognitive Systems: Human Cognitive Models in Systems Design.* Erlbaum, Mahwah, NJ. chapter AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue.

Hays, D. 1964. Dependency Theory: A Formalism and Some Observations. *Languages*, 40: 511–525.

Kozareva, Z., and Montoyo, A. 2006. *Lecture Notes in Artificial Intelligence: Proceedings of the 5th International Conference on Natural Language Processing (FinTAL 2006)*. chapter Paraphrase Identification on the basis of Supervised Machine Learning Techniques.

Ibrahim, A.; Katz B.; and Lin, J. 2003. Extracting Structural Paraphrases from Aligned Monolingual Corpora. in *Proceeding of the Second International Workshop on Paraphrasing*, (ACL 2003).

Iordanskaja, L.; Kittredge, R.; and Polgere, A. 1991. Natural Language Generation in Artificial Intelligence and Computational Linguistics. *Lexical selection and paraphrase in a meaning-text generation model*, Kluwer Academic.

Lin, D. 1993. Principle-Based Parsing Without Overgeneration. In *Proceedings of ACL*, 112–120, Columbus, OH.

Lin, D. 1995. A Dependency-based Method for Evaluating Broad-coverage Parsers. In *Proceedings of IJCAI-95*.

Lintean, M.; Rus, V.; and Graesser, A. 2008. Using Dependency Relations to Decide Paraphrasing In *Proceedings of the Society for Text and Discourse Conference 2008*.

Marneffe, M. C; MacCartney, B.; and Manning, C. D. 2006. Generating Typed Dependency Parsers from Phrase Structure Parses. In *Proceeding of LREC 2006*.

McNamara, D.S.; Boonthum, C.; Levinstein, I. B.; and Millis, K. 2007. *Handbook of Latent Semantic Analysis.* Erlbaum, Mahwah, NJ. chapter Evaluating self-explanations in iSTART: comparing word-based and LSA algorithms, 227–241.

Miller, G. 1995 WordNet: A Lexical Database of English. Communications of the ACM, v.38 n.11, p.39-41.

Patwardhan, S.; Banerjee, S.; and Pedersen, T. 2003. Using Measures of Semantic Relatedness for Word Sense Disambiguation. in *Proceeding of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February.

Qiu, L.; Kan M. Y.; and Chua T. S. 2006. Paraphrase Recognition via Dissimilarity Significance Classification. In *Proceeding of EMNLP*, Sydney 2006.

Rus, V.; McCarthy, P. M.; Lintean, M.; McNamara, D. S.; and Graesser, A. C. 2008. Paraphrase Identification with Lexico-Syntactic Graph Subsumption. In *Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS-2008).*

Rus, V.; McCarthy, P. M.; McNamara, D. S.; and Graesser, A. C. 2008. A Study of Textual Entailment. *International Journal of Artificial Intelligence Tools*, August 2008.

Wu, D. 2005. Recognizing Paraphrases and Textual Entailnment using Inversion Transduction Grammars. in *Proceeding of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailnment*, Ann Arbor, MI.

Zhang, Y., and Patrick, J. 2005. Paraphrase Identification by Text Canonicalization In *Proceedings of the Australasian Language Technology Workshop 2005*, 160–166.